Article
ECORFAN Journal-Taiwan
December 2018 Vol.2 No.4 11-21

Analysis of performance of AI planning algorithms and hybrid methodology in a problem of generation of learning paths

Análisis del rendimiento de los algoritmos de planificación de IA y la metodología híbrida en un problema de generación de rutas de aprendizaje

MAYA-PADRÓN, Cristina*†, SANCHEZ-NIGENDA, Romeo

Universidad Politécnica de García: Porfirio Diaz 100, Centro, García, N.L., México |
Universidad Autónoma de Nuevo León: Pedro de Alba s/n, San Nicolás de los Garza, N.L., México

ID 1st Author: *Cristina, Maya-Padrón* / **ORC ID:** 0000-0001-9982-9991, **Researcher ID Thomson**: X-3082-2018, **CVU CONACYT ID**: 386884

ID 1st Coauthor: *Romeo, Sanchez-Nigenda* / **ORC ID:** 0000-0001-7272-3759, **Researcher ID Thomson**: A-3285-2012, **CVU CONACYT ID**: 87242

Received July 30, 2018; Accepted December 30, 2018

Abstract

In this paper we present an analysis of the performance of two Artificial Intelligence pplanning algorithms, SGPLAN and LPG, in a problem of generation of learning paths (GLP). Likewise, two models were developed to represent this problem: a) as a model of Artificial Intelligence Planning, with the planning domain definition language (PDDL), which uses the planning algorithms SGPLAN and LPG for its solution; and b) as a mathematical model. It also presents a hybrid methodology of solution in which both planning and mathematical models are combined. In the experimentation the performance of the planning algorithms is evaluated to obtain solutions (plans) comparing the results obtained by both models. And finally, the performance of the planning algorithms is observed when modifying the planning models with information of solutions obtained with the mathematical model (hybrid method). We hope that the results obtained in this research serve to highlight the benefits of using AI planning and the planning algorithms SGPLAN and LPG for their solution. As well as showing the opportunity areas of such algorithms.

Artificial intelligence planning, Planning algorithm, Mathematical modeling, Hybrid methodology, Learning paths

Resumen

En este trabajo se presenta un análisis del desempeño de dos algoritmos de planificación de Inteligencia Artificial, SGPLAN y LPG, en un problema de generación de rutas de aprendizaje (GLP). Asimismo, se desarrollaron dos modelos para representar dicho problema: a) como un modelo de Planificación de Inteligencia Artificial, con el lenguaje de definición de dominios de planificación (PDDL), que utiliza para su solución los algoritmos de planificación SGPLAN y LPG; y b) como un modelo matemático. Se presenta además una metodología híbrida de solución en donde se combinan ambos modelos de planificación y matemático. En la experimentación se evalúa el desempeño de los algoritmos de planificación para obtener soluciones (planes) comparando los resultados obtenidos por ambos modelos. Y finalmente, se observa el desempeño de los algoritmos de planificación al modificar los modelos de planificación con información de soluciones obtenidas con el modelo matemático (método híbrido). Esperamos que los resultados obtenidos en esta investigación sirvan para resaltar los beneficios de utilizar la planificación de IA y los algoritmos de planificación SGPLAN y LPG para su solución. Así como mostrar las áreas de oportunidad de dichos algoritmos.

Planificación de Inteligencia Artificial, Algoritmos de planificación, Modelación matemática, Metodología híbrida, Rutas de aprendizaje

Citation: MAYA-PADRÓN, Cristina, SANCHEZ-NIGENDA, Romeo. Analysis of performance of AI planning algorithms and hybrid methodology in a problem of generation of learning paths. ECORFAN Journal-Taiwan. 2018, 2-4: 11-21.

* Correspondence to Author (email: cristina.maya@upg.com.mx)
† Researcher contributing first author.

Introduction

In this paper, an evaluation of the performance of two planning algorithms is carried out to solve a problem of generation of learning paths.

The planning algorithms are used to solve Artificial Intelligence Planning models. Commonly they are called "*independent planners of the domain*", this is because regardless of the planning model in question, to be an educational model like ours, of scheduling, purchasing, manufacturing, etc., they can obtain a solution (a plan) to said model.

Planners are special-purpose algorithms that use a formal planning language such as PDDL (planning domain definition language), with well-defined syntax, semantics, and demonstration theory (Russell & Norvig, 2004).

Two planning algorithms are considered: SGPLAN and LPG, which were selected for their high performance in the International Planning Competition, bi-annual event organized in the framework of The International Conference on Automated Planning and Scheduling (ICAPS), which is the premier forum for exchanging news and research results on theory and applications of intelligent planning and scheduling technology (ICAPS, 2018).

The planning model selected to evaluate the performance of the planning algorithms is the *problem of generation of learning paths (GLP)* (Sanchez, et.al, 2017).

The *problem of generation of learning paths* can be considered as follows: Considering that it has an academic program, which contains the subjects that the student must study, and that in turn each subject is composed of a set of themes or learning units, it is assumed that there are various learning activities modeled by the instructor/educator, so that the student can perform to each learning objective.

In this way, it can generate an ordered sequence of learning activities for the student to allowed him to minimize the total time he has dedicated to his activities, considering a utility (score) for each approved learning objective. It also considers, the precedence of learning activities and the activities considered as mandatory.

Therefore, we see that it is a complex problem, since activities must be selected that comply with the objective function that is to minimize the total time, complying with a set of restrictions such as mandatory activities, precedence in activities, considering an evaluation or score of approval defined by the user.

This problem (GLP) was modeled in two ways: like an AI planning model, using PDDL and as a mathematical model. It is worth mentioning that the order of activities is not considered in the mathematical model. To represent the precedence of activities we do the following: if a learning activity is selected and it has an activity that precedes it, the mathematical model is forced to select both activities. This is to ensure that in some way there is a sequence, as it would be the planning model, even if the all activities do not go in order.

Next, the methodology uses, the approach of both developed models, results, conclusions and references will be described.

Methodology

With regard to artificial intelligence planning

The methodology used in this work is as follows: The problem of generation of learning paths is modeled as an artificial intelligence planning model, this is done by the PDLL planning domain definition language (Fox & Long, 2003). The *planners* are used to obtain *solutions* from the planning models, and finally, as a result of the planning process, the plans are obtained, which are the learning paths. In Figure 1 you can see the complete picture of the *planning process*.

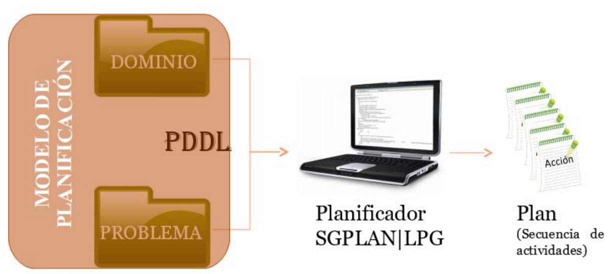

Figure 1 Planning process

In order to model the problem of generation of learning paths as a planning model, we use the PDDL 2.1 planning domain definition language.

PDDL is a language centered on actions inspired by the formulations "Strips" of planning problems. This is a standardization of the syntax to express actions using preconditions and post-conditions to describe the applicability and effects of the actions. The syntax is inspired by Lisp (Winston & Horn, 1989) (acronym for LISt Processing), so much of the structure of the domain description is a list as Lisp of the expressions in parentheses.

A *planning model* in PDDL is organized into two main parts: the *planning domain* and the *planning problem* (see the previous Figure 1). The *domain* is the construction of the model and the *problem* is an instance to solve of that domain. However, it is common to use the term planning domain to refer to the planning model itself. The following describes how the planning model is organized in PDDL:

1. ***Planning domain****:*

Which describes the rules of action and is composed of:

1.1. *Predicates*: These represent relationships between the objects. They help us describe a problem in the real world trying to represent the concepts of our problem through relationships between the objects that make it up.
1.2. *Fluents*: Functions that allow us to handle numerical values.
1.3. *Actions/Operators*: Ways to change the state of the world.

2. ***Planning problem***

It describes the state of the surrounding world and the goals/objectives, as well as the metrics to be optimized. It is composed of:

2.1. *Objects*: The things in the world that are of our interest.
2.2. *Initial state*: The state of the world in which you start. That is, the starting point of the search.
2.3. *Specification Objective*: Check if the current status corresponds to a solution to the problem.
2.4. *Metric to be optimized*: In the case of planning models that handle Fluents, select the actions that comply with the metric to be optimized.

The actions: they are part of the planning domain in PDDL (section 1.3 above) are the ways to change the state of the world. Next, its components are mentioned:

– *Specification of the action*: it is what the agent actually returns to the environment in order to proceed to do something. When it is inside the planner it serves as the name of the action.

– *The precondition*: is a conjunction of atoms (positive literals) or functions (fluents) that say what must exist (be true) before the operator can apply it.

– *The effect of an operator*: is a conjunction of literal (positive or negative) or functions (fluents) that tells how the situation changes when applying the operator.

Once the *planning model* has been developed as it is described above, the *planners* are used to solve it. We call it the *solution* of the result of the *AI planning process*. The term *solution* according to (Russell & Norvig, 1996) is a *plan* that an agent can execute and guarantees the achievement of the goal. That is, a sequence of *actions* that are executed in the *initial state*, and results in a *final state* that satisfies the objective.

Next, we describe the selected planners to evaluate:

SGPLAN: is a planner that was in the first place in the IPC of 2006 in the deterministic part of the competition. SGPlan partitioned a large planning problem into subproblems, each with its own sub-objectives. The version we use for our experimentation is SGPlan-5.

LPG: This planner has participated in the IPC and was awarded as the best automated planner in 2003 and later in the IPC 2004 was awarded as the best performance. LPG is a stochastic planner, based on searches in the forward state space. It is recommended for domains that have numerical quantities and durations like our GLP problem. The version we use for our experimentation is LPG-td-1.0.

With regard to mathematical modeling

In the scientific approach of decision making, the use of one or more mathematical models is required. These are mathematical representations of real situations that could be used to make better decisions, or simply to better understand the current situation.

A model of this type dictates behavior for an organization that will allow you to achieve better your goal(s).

The elements of a mathematical model are:

– *Function (s) objective*: In most models there is a function that we want to maximize or minimize.

– *Variables decision*: They are variables whose values are under our control and influence the performance of the system.

– *Restrictions*: In most situations, only certain values of the decision variables are possible.

There are different mathematical models that can be made, this is according to the nature of the decision variables and how the objective function and restrictions are defined. Among the different types of mathematical models are: linear models, non-linear models, integer models, non-integer models, binary models and mixed-integer linear programming (MILP) models.

Among these type of mathematical models, we are interested in the MILP model: If in a linear model only some variables are restricted to integers, then we have a mixed integer linear model.

With regard to combination of both methods

The mathematical model provides us with exact solutions regarding the selection of learning activities that minimize the total time, which is the objective function defined for our problem, but does not consider the ordering of learning activities. The planning algorithm seeks to do both, however, it throws very large GAPs, that is, the selection of activities is far above an optimal or exact solution,

Taking into account the above, we consider the solutions of the mathematical model and include them in the planning model (see figure 2). This aims to make it easier for the planning algorithm to obtain better solutions, since it will not make the selection of activities that optimize the objective function. Therefore that task will have been performed by the mathematical model. So the algorithms will consider the activities already selected by the mathematical model and will perform the ordering. The advantage of this hybrid methodology is to obtain better solutions, with respect to the GAP.

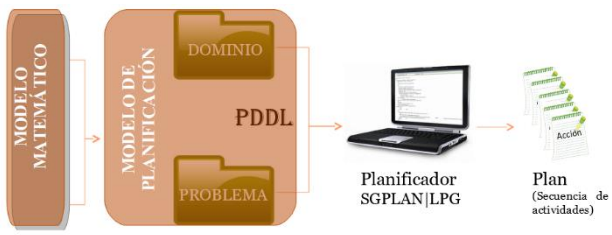

Figure 2 Hybrid methodology: planning process with the mathematical model

Approach of the developed models

We can raise the problem of generation of learning routes as one that automatically generates an ordered sequence of activities that allow optimizing a metric. Figure 3 shows graphically the GLP problem, where you have a subject, which has a number of themes or units of learning, these have specific learning objectives that are the sub-themes. Each subtheme has learning activities that must be done to meet that specific objective. The learning activities have duration, score and an associated resource, this is the time it takes to perform the activity, the score or utility obtained when doing it, which we assume is proportional to the duration and an associated resource such as a book, computer, etc., necessary to carry out the activity.

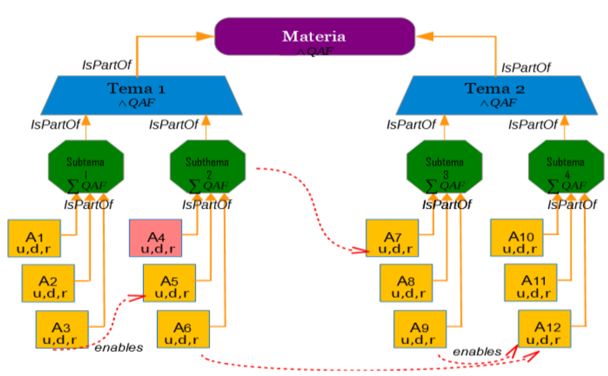

Figure 3 Graphical representation of the GLP problem

ISSN-On line: 2524-2121

In addition to the above, you can see in figure 3 the red dashed arrows between the activities, this represents the precedence relations between the learning activities. For example, activity A5 is a reading of chapter 2 of a given book and activity A3 is the chapter1 that gives the introduction. So that there may be relationships of only one previous activity (1:1) or two previous (2:1).

Activity A4 is marked with a different color because it is a mandatory activity, that is, this activity must be part of the sequence of activities to be carried out. An example of this type of activity is an evaluation test.

$\sum QAF$: Represents the sum of the score or utility for each subtheme. This is important because this helps us to define the activities to be carried out according to what qualification or utility is expected of each subtheme.

$\wedge QAF$: This indicates that all the subthemes and all the themes of a subject must be done, to consider that it is fulfilled.

Planning Model

```
(:durative-action enroll-subject Materia1
:parameters (?s - student)
:duration (= ?duration 0)
:condition (and
(at start (available-subject Materia1 ?s))
(at start (not-approved Materia1 ?s))
(at start (<(credits-subject Materia1)(available-credits ?s)))
)
:effect (and
(at end (enrollment ?s Materia1))
(at end (decrease (available-credits ?s)(credits-subject Materia1)))
(at end (not (available-subject Materia1 ?s)))
))
```

Figure 4 Action to enroll a subject

Figure 4 shows the action of enroll a subject for a student. The characteristics of this action are: it has as parameters the student; it has no duration; preconditions: the subject must be available, the student must not have previously approved the subject, the credits of the subject must be less than what a student can take in a period (This is for control of the number of subjects to take, for example, a student wants to take 10 subjects in a period and the allowed is only 5 or 6, according to the credits that each subject has).

As a result of carrying out this action (effect of the action): the student is enrolled in the subject, so that he can perform any activity of it; the available credits decrease; the subject is no longer available to re-enroll, since it is the one that is studying.

Figure 5 shows the action that represents those activities that have no relation of precedence with any other activity, an example of this type of activities is activity A1 of figure 3 above. The characteristics of this action are: it has as parameters the student, the learning activity, the subtheme, the theme and subject; the action duration is according to each activity. The preconditions are: that the student is not busy; the student has enrolled the subject; the student has not done that same activity before; the activity belongs to the subtheme, this is part of the theme and the enrolled subject; the type of resource of the activity is available; the activity has no relation of precedence, and finally, that the score or utility of the learning activity is not greater than what is defined by the user as utility or maximum score per subtheme (prevents more activities from being performed in an unnecessary way).

The effect of performing the action: the student appears as busy and the necessary resource for the activity is not available during the time of the activity; increases the score or utility in the subtheme, (representing $\sum QAF$). At the end of the action, the student is available for a new activity, the utility or score of the subtheme decreases, and the activity is marked as done.

```
(:durative-action CHOOSE-LA-nothasreqs
:parameters (?s - student ?oa - LA ?subt - subtheme ?t - Theme ?subj - subject ?eq
- resource)
:duration (= ?duration (DurationLA ?oa))
:condition (and
(at start (free ?s))
(at start (enrollment ?s ?subj))
(at start (not-done-LA ?oa ?subj ?s))
(at start (isPartOfSubtheme ?oa ?subt))
(at start (isPartOfTheme ?subt ?t))
(at start (isPartOfSubject ?t ?subj))
(at start (KindResourceLO ?oa ?eq))
(at start (> (quantity-resource ?eq) 0))
(at start (not-has-reqs ?oa))
(at start (> (maxgrade-subtheme ?subt)(valueLA ?oa)))
)
:effect (and
(at start (not(free ?s)))
(at start (decrease (quantity-resource ?eq) 1))
(at end (increase (quantity-resource ?eq) 1))
```

```
(at end (not (not-done-LA ?oa ?subj ?s)))
(at end (increase (score ?subt ?s) (valueLA ?oa)))
(at end (free ?s))
(at end (decrease (maxgrade-subtheme ?subt)(valueLA ?oa)))
(at end (done ?oa))
))
```

Figure 5 Action to select activities that have no precedence relationship

For actions where activity precedence is taken, the following conditions are added in the precondition part of the action (see figure 6).

```
(at start (has-reqs ?oa ?req))
(at start (done ?req))
```

Figure 6 Sentences in the pre-condition of action of activities with precedence

Likewise, if you want to represent the precedence of two learning activities, it is as follows (see figure 7)

```
(at start (not(= ?req1 ?req2)))
(at start (done ?req1))
(at start (done ?req2))
(at start (has-multiple-reqs ?oa ?req1))
(at start (has-multiple-reqs ?oa ?req2))
```

Figure 7 Sentences in the pre-condition of the action of activities with precedence to two activities

To represent $\wedge QAF$, in the planning model it is divided by actions for themes and subject. To ensure that all subthemes of a theme are carried out, it is defined as follows (see figure 8).

```
(:durative-action PASS-Theme-Tema1 Materia1
:parameters (?s - student)
:duration (= ?duration 0)
:condition (and
(at start (enrollment ?s Materia1))
(at start (>= (score Subtema1 ?s)(mingrade Materia1)))
(at start (>= (score Subtema2 ?s)(mingrade Materia1)))
)
:effect (and
(at end (done-Theme Tema1 Materia1 ?s))
))
```

Figure 8 Action to complete a theme

This action has the following characteristics: it has as parameter the student; it has no duration; the preconditions are: that the student is enrolled in the subject, and that what has accumulated score for each subtheme is defined by the user (for each subtheme).

The effect of the action: the subject is marked as completed (if there are more themes in a subject, an action is made for each one). In the case of mandatory activities such as activity A4 in Figure 3, a predicate is indicated in the action to approve the theme (see figure 9).

```
(at end (done ?oa))s
```

Figure 9 Sentence for mandatory activities

Like the previous action, to indicate that all the themes have been done by subject, it is indicated in an independent action (see figure 10). This action is similar to the previous one, only that in this action the preconditions are: that the themes that comprise the subject have been completed. The effect of performing the action: is that the subject is completed. As in the theme, there is an action for each modeled subject.

```
(:durative-action PASS-Materia1
:parameters (?s - student)
:duration (= ?duration 0)
:condition (and
(at start (enrollment ?s Materia1))
(at start (done-Theme Tema1 Materia1 ?s))
)
:effect
(at end (done-subject-LA Materia1 ?s))
)
```

Figure 10 Action to approve subject

As it was already mentioned, the problem file of the planning model is an instance of the GLP problem, which indicates the student's situation, the learning activities that each subtheme has, the duration, score or utility of each activity, as well as the type of resource associated to each one of them, and the amount of them. It also shows the hierarchical relationship of which subtheme belongs to which theme and subject. Each problem file will be made according to the subject matter and the current situation of each student.

In addition to the previous information in the problem file, the objective and the metrics to be optimized are defined, this can be seen in figure 11, where it is indicated that the objective is for the student to approve a certain subject (or several), and the metric to optimize is to minimize the total time.

December 2018 Vol.2 No.4 11-21

```
(:goal (and
(pass-degree Materia1 student1)
))
(:metric minimize (total-time))
)
```

Figure 11 Objective and metrics to be optimized

Mathematical Model

We developed a mixed integer linear programming (MILP) model. In it we relax the ordering restrictions of the activities. It does not consider the hierarchy of the network, since the intention is that it makes the selection of those activities of some subtheme that minimize the total time. This considers the activities that are mandatory, and if there is a relationship of precedence between activities, what it does is to ensure that both activities are included in the plan if one of them is selected by the mathematical model.

Assumptions

- The activities are done only once.
- In accordance with the metric to be optimized there will be activities that are not selected.
- All activities have a duration and utility (score).
- It is not considered the sequencing (order) of activities.

Parameters:

u_{ij} : Utility value of the learning activity *i* del subtheme *j*.
d_{ij} : Duration of the learning activity *i* del subtheme *j*.
$Kmin$: Minimum passing grade that the educational plans will have to meet in each subtheme.
$Kmax$: Maximum grade that a student can obtain per subtheme.
Sets:
M : Mandatory learning activities set, such that $m_{ij} \in M$ if the learning activity *i* of subtheme *j* is mandatory.
W : Binary matrix of size nxn learning activities, where $w_{ii'} = 1$ if learning activity *i* enable the learning activity *i'*, and $w_{ii'} = 0$ in other case.

Decision variable and auxiliary variable

$x_{ij} = \mathbf{1}$, if the learning activity *i* of subtheme *j* is completed. 0 in other case.

y_j: Auxiliary variable that represents the cumulative score or utility in the subtheme *j*.

Nature of the variables:

$i \in \mathbb{N}, i = 1, \dots, n$ learning activities
$j \in \mathbb{N}, j = 1, \dots, a$ subthemes

Mathematical model

Objective Function

$$\min z = \sum_{i=1}^{n} \sum_{j=1}^{a} d_{ij} x_{ij} \quad (1)$$

Restrictions:

$$\text{j} = 1,2,\dots,a \sum_{i=1}^{n} u_{ij} x_{ij} \geq Kmin \quad (2)$$

$$\text{j} = 1,2,\dots,a \sum_{i=1}^{n} u_{ij} x_{ij} \leq Kmax \quad (3)$$

$$\text{j} = 1,2,\dots,a y_j = \sum_{i=1}^{n} u_{ij} x_{ij} \quad (4)$$

$$j, j' = 1,2,\dots,a \text{i}, \text{i}' = 1,2,\dots,n; \quad (5)$$
$$x_{i'j'} \leq w_{ii'} x_{ij}$$

$$\forall m_{ij} \in M$$
$$\text{x}_{\text{ij}} = 1 \quad (6)$$

$$j = 1,2,\dots,a i = 1,2,\dots,n; \text{x}_{\text{ij}} \in \{0,1\} \quad (7)$$

$$\text{j} = 1,2,\dots,a y_j \geq 0 \quad (8)$$

The objective function (1) is to minimize the total time of the activities to be performed. The set of restrictions (2) ensures that the sum of the utility value of the selected learning activities is greater than or equal to a minimum passing score per subtheme. The restrictions (3) are similar to the previous ones, but for the upper limit of the rating for each subtopic. This set of restrictions may seem trivial for the objective function z, since in it we consider time, but it is important to bear in mind that the set of activities selected for the plan consider that they ensure a score or utility per subtheme. The set of restrictions (4) provides information on the cumulative score obtained in each subtheme.

The restrictions (5) guarantee that each relation of precedence (or enabling) between the learning activities are considered in the solution, that is, if there is a relation of precedence between two activities, although it is not considered which one occurs first and which one after, it is considered that if one is selected to be in the plan, the other is also. The restrictions (6) ensure that each mandatory learning activity is included in the plan. Finally, the restrictions (7, 8) establish the nature of the variables.

Results

We divided experimentation into two sections. In the first section, the comparison of both solutions is carried out: those generated by the planning algorithms and the solutions obtained by the mathematical model. In the second section the results of mixing the solutions of the mathematical model to the planning models are observed.

Comparison of both solutions

A generator was developed in Ansi C which generated 450 instances of the planning model and 450 of the mathematical model. The design of the instances considers three classes of different models, represented in table 1.

	Subjects	Themes	Subthemes	Learning Activities
Small	1	1	2	5
Medium	1	3	4	5
Large	1	5	6	5

Table 1 Classes of the instances

Model classes vary in the number of learning tasks they represent. For this we established a single subject and we varied the number of themes, subthemes and learning activities. We consider that a single student is modeled, the resources are unlimited, and the objective function is to minimize the total time, as defined in the mathematical model (1).

The characteristics of the instances can be seen in Table 2. Where we can see: 1) the percentage of precedence relationships presented in the model, being 0%, (has no relation of precedence), 20% and 80% of the total of activities.

Characteristics of the instances			
% Precedence relationships	% Type of precedence relationships	% Mandatory activities	Range of values in the utility
0%	0%	[0%, 10%, 20%]	[UF, RD]
20%	1:1 (20%) 2:1 (80%)	[0%, 10%, 20%]	[UF, RD]
	1:1 (80%) 2:1 (20%)	[0%, 10%, 20%]	[UF, RD]
80%	1:1 (20%) 2:1 (80%)	[0%, 10%, 20%]	[UF, RD]
	1:1 (80%) 2:1 (20%)	[0%, 10%, 20%]	[UF, RD]

Table 2 Characteristics of the instances

2) We consider that if there are relations of precedence they could be: to a single activity (1:1) or to two activities (2:1). Varying between 20% and 80% for each type

3) We distribute the mandatory activities in 0%, 10% y 20%.

4) We have two types of range of values assigned to the learning activities as utility: Uniforms (UF) and Random (RD). In UF the assigned value is + - 5 points of the average. The average is calculated as follows: the maximum grade (100) that could have an activity, among the amount of learning activities of a subtheme, for example, if there are 5 activities is 100/5 = 20, then each activity will be between 15 and 25 points each. It must be ensured that the sum of all is 100. For the RD values, random numbers are generated in a range of [10.70], having a wide utility between each activity. It must be ensured that at least half of the activities add up to 70 so that unfeasible instances are not generated.

Five different instances of the model were generated per test case for both models, it was established that the utility per subtheme is greater than or equal to 70.

The General Algebraic Modeling System, which is a high-level modeling system for mathematical programming and optimization (GAMS, 2018), is used to optimally solve mathematical programming models.

The resolution of the mathematical programming models provides optimal solutions in all generated instances. This in terms of the selection of activities that minimize the total time, defined in the objective function.

To identify the difference between the optimal solutions generated by the mathematical model and the planners' solutions, we calculate their GAP (difference with respect to the optimum). Figure 11 shows the GAP, organized by model classes and planners, and figure 12 shows all instances of the model (450) by planner.

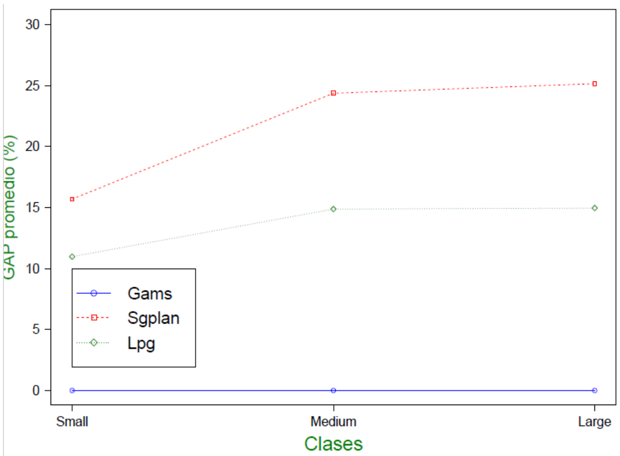

Figure 11 GAP of SGPLAN and LPG per class

In Figure 11 it is easy to observe that on average, SGPLAN has a larger GAP compared to LPG. Having a GAP between 15% and 25%, and LPG between 10% and 15% on average per class. It can also be observed that both algorithms increase their GAP by going from the small class to the large one.

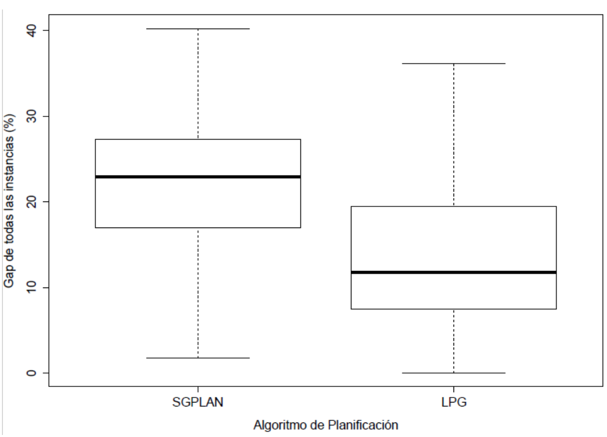

Figure 12 GAP of all instances

We show the average GAP for all instances (450) in Figure 12, where we can see that on average SGPLAN has a GAP of 21.66% and 13.59% for LPG. Regarding the percentage of optimal solutions found by the planners, we have that SGPLAN could find the optimal solution in 4.89% and LPG in 8.22%, being, in both cases, instances of the small class.

On the other hand, the percentage of instances resolved by both the mathematical model and the planning algorithms is high. GAMS resolved 100% of the instances, SGPLAN 97.78% and LPG 99.11%; being all unresolved instances of the medium and large classes.

With respect to the running time, we can see in Figure 13 that the time taken to solve the models is negligible, considering that all the solutions are in microseconds. However, we can see that GAMS takes more time on average to solve the instances, LPG is the second longest, considering that in large instances the time shoots from less than 20 to almost 100 ms. SGPLAN was the one that took less time to solve.

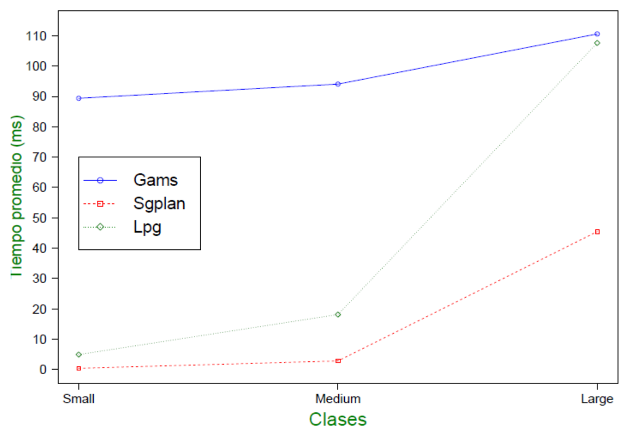

Figure 13 Running time in microseconds

It is interesting to see in Figure 14 the average cumulative utility per subtheme obtained by the planning algorithms and the mathematical model. The data, which is in ascending order by class, shows that although the planning algorithms come with a large GAP, their learning paths guarantee better grades for a student. It is observed that SGPLAN gives utilities around 85, LPG around 80 and GAMS around 75.

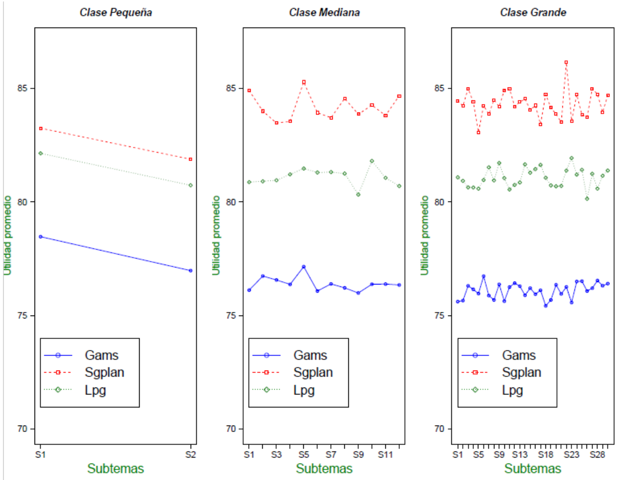

Figure 14 Average accumulated utility by subtheme

In conclusion we can say that LPG presents a better performance in terms of the quality of the solution with respect to SGPLAN, since it does not only have a lower GAP but also resolved almost 100% of the instances with a greater number of optimal solutions. However, in terms of computational time and the accumulated utility per subtheme SGPLAN is better.

Considering mathematical model solutions as entry into the planning model

The mathematical model provides us with exact solutions regarding the selection of learning activities that minimize the total time (objective function), without considering the sequence (ordering) of the learning activities. The planning algorithms seek to do both, but it throws very large GAPs.

For this reason we consider the solutions of the mathematical model and include them in the planning models so that the planning algorithms perform the ordering and obtain better solutions with respect to the GAP.

For experimentation we consider the same instances of the previous section modifying the planning models with the solution of the mathematical model. We ask to solve with the planning algorithms SGPLAN and LPG already indicated above.

The modification in the planning models is to leave in these models only the learning activities selected by the mathematical model.

There are two ways in which we can modify the goal in the planning model: the first is to put as objective: "approve the subject", leaving the calculation of the metrics equal. The other way is to establish as a goal: to "approve each of the selected learning activities" by the mathematical model, and to remove the calculation of the metrics, that is, the one that calculates the number of activities that can be carried out per subtheme.

Objective: Pass the subject

With respect to the percentage of resolved instances we have that, SGPLAN and LPG found a solution of 98.44% of the instances. This is not very different from the percentage of previous experimentation (without exact solutions). However, the solution time is shorter in both planning algorithms. Figure 15 shows the solution times of both planners in ms. The results are shown before (legend SGPLAN, LPG) and after mixing both solutions (legends SGPLAN-M, LPG-M). You can see that the solution times improve with respect to previous experimentation.

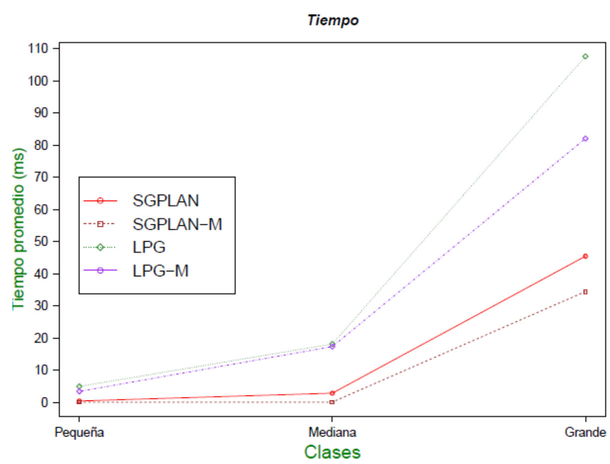

Figure 15 Average time of solution of both planning algorithms with objective of pass the subject

Objective: To do each activity

With respect to the percentage of resolved instances we have that, SGPLAN and LPG found 100% solution of the instances. Presenting a higher percentage of resolved instances both in the experimentation without exact solutions, and in the experimentation with the objective of approving the subject.

ISSN-On line: 2524-2121

In the same way, it is observed that the solution time decreases in both planning algorithms. Figure 16 shows the solution times of both planners in ms. The results are shown like before (legend SGPLAN, LPG) and after mixing both solutions (legends SGPLAN-M, LPG-M). You can see that the solution times improve with respect to the previous experimentation, even decrease with respect to the experimentation with the objective of pass the subject. SGPLAN had a solution time reduction of 38.70% in the medium class and 49.54% in the large class. While LPG had a reduction of 33.78% in the medium class and 39.66% in the large class. This compared to the results without exact solutions.

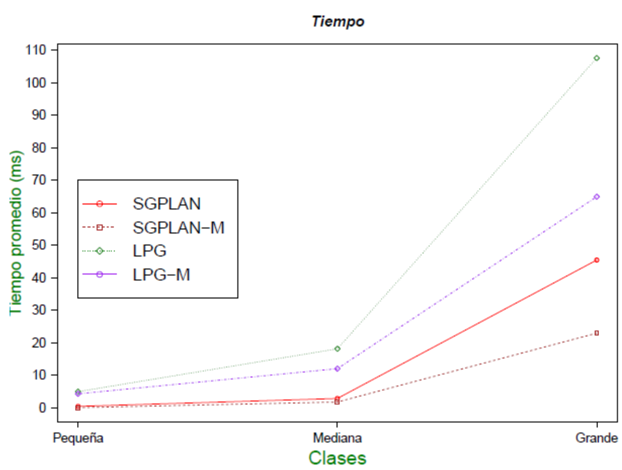

Figure 16 Average time of solution of both planning algorithms with the objective of to do each activity

In conclusion we can say that there is a benefit when the solutions of the mathematical model are included in the planning models. Since as we could observe in the results of the experimentation, the planning algorithms have presented difficulty in the selection of activities according to the established metrics.

Conclusions

In this article, two different models were presented to generate learning paths for students: Mathematical Modeling and Artificial Intelligence Planning. Experimentation was conducted to observe the performance of two planning algorithms: SGPLAN and LPG. We used a hybrid method in which we linked the mathematical programming and the AI planning to improve the quality of the solutions obtained of the planners. Noting that when both methodologies are linked, better solutions are obtained.

As part of future work we are considering to work in the development of a planning algorithm that adequately calculates the metrics of the actions and generates optimal solutions in the obtained plans. In addition to working on a user interface that allows interaction with students and generate a graphically the plans obtained.

References

Fox, Maria & Long, Derek. (2003). PDDL 2.1: An Extension to PDDL for Expressing Temporal. Journal of Artificial Intelligence Research.

GAMS. (2018). Home Page General Algebraic Modeling System (GAMS). Sitio web: https://www.gams.com/

Hsu, Chih-Wei & W.Wah, Benjamin. (2008). The SGPlan Planning System in IPC-6. Association for the Advancement of Artificial Intelligence.

ICAPS. (2018). Home Page ICAPS. 2018, de ICAPS Sitio web: http://ipc.icaps-conference.org/

Russell, Stuart & Norvig, Peter. (1996). Inteligencia Artificial. Un enfoque Moderno. México, D.F.: Prentice Hall Hispanoamérica, S.A.

Russell, Stuart & Norvig, Peter. (2004). Inteligencia Artificial. Un enfoque Moderno. Madrid: Pearson Educación.

Sanchez, R., Maya, C. & Martinez, I. (2017). Design and evaluation of planning and mathematical models for generating learning paths. Computational Intelligence, pp.1-18

Winston, Patrick & Horn, Berthold. (1989). LISP. Estados Unidos de América: Addison-Wesley.